Neuro-Fuzzy Recognition of Overlapping Handwritten Text between Adjacent Lines of Text using Soft Computing

Raj Sheth, Ketan Patil, Niharika Thakur, Prof. K.T.Talele
Sardar Patel Institute of Technology,
Mumbai 400058, India

***Abstract*—The research aims at developing an efficient handwriting recognition system used for detecting overlapped handwritten characters between adjacent lines of text. It focuses mainly on detection of the overlapped region and separating the overlapped handwritten lower case non-cursive English alphabets. Considering only the most probable cases for overlapping characters between adjacent lines of text, the recognition takes place by a novel technique where only the non-overlapped portion is considered for processing, and by reasoning which uses the knowledge of their structure, we digitize and detect the individual overlapped characters. The overlapped characters were successfully recognized with an accuracy of 85.92%. The non-overlapping characters are fed to the feed forward back propagation neural network where detection takes place. Here, diagonal based feature extraction is used.**

***Index Terms*—Handwritting Recognition, Image Processing, Pattern Recognition, Neural Networks, Feature Extraction**

I. Introduction

Handwriting recognition is the ability of any system to read, recognize and analyze information. Numerous attempts have been made to develop an efficient handwriting recognition system which generally incorporates the method optical character recognition. An intelligent handwriting system might perform the functions of formatting, character segmentation and also finding the words with almost 100% accuracy.

However, among all the successful attempts in related research over the past two decades, the research on the recognition of overlapped characters has never evolved to significant heights [1] [3] [5] [6].

Handwriting is natural and differs from person to person. Here, even the best of the handwriting recognition systems have not been successful in distinguishing between overlapped characters in the handwritten text [1] [3].

Overlapping of characters generally take place in two major instances. One, overlapping of characters placed adjacent to each other or the horizontal overlap, and two, overlapping in adjacent lines or the vertical overlap.

The answer to this dilemma is our research on various methods for developing an efficient handwriting recognition system which mainly focuses on separating hand-written overlapped characters between adjacent lines and then digitizing it.

II. Different Types of Overlapping Cases

In English language, the most probable cases for overlapping will only be those characters which either tread vertically upwards or a vertical downwards. Hence, two categories of characters have been defined, ascenders and descenders. Ascenders are the alphabets which rise above a horizontal line while writing as shown in the figure 2, while descenders are those which move downward while writing as shown in the figure 1. The only possible cases of ascenders and descenders in English alphabets are given below which are found by observation.

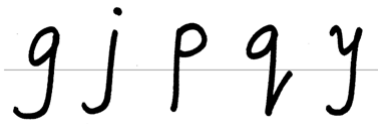

Fig. 1. Descenders or Alphabets treading vertically downwards

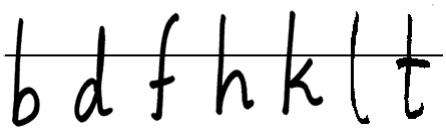

Fig. 2. Ascenders or Alphabets treading vertically upwards

III. Proposed System

In this section, the proposed recognition system as shown below 3 is described.

A. Image Acquisition

The recognition system acquires a scanned image as an input image as shown in the figure 4. The images used here are in black and white form in any format such as JPEG, BMT, BMP, etc. This image is acquired is forwarded to the subsequent blocks for further processing.

B. Noise Removal

The image resulting from the scanning process may contain a certain amount of noise. Depending on the resolution on the scanner and the success of the applied technique for thresholding, the characters may be smeared or broken. Some of these defects, which may later cause poor recognition rates, can be eliminated by using a preprocessor to smooth

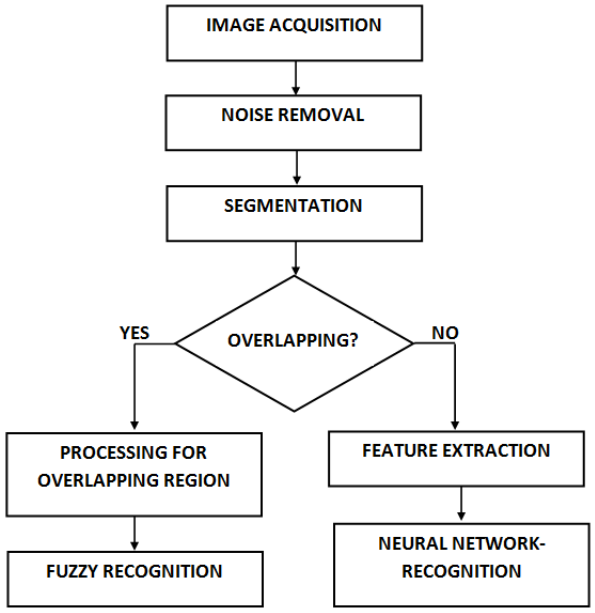

Fig. 3. Block Diagram of the Proposed System

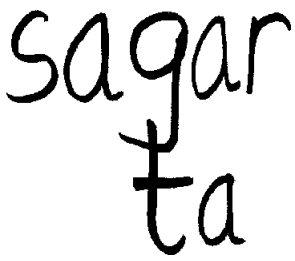

Fig. 4. Input Image as word

the digitized characters. The smoothing implies both filling and thinning. Filling eliminates small breaks, gaps and holes in the digitized characters, while thinning reduces the width of the line. In addition to smoothing, preprocessing usually includes normalization. The normalization is applied to obtain characters of uniform character width and size. [9]

To eliminate noise from the input image, the connected pixels of very small area are deleted from the image by making use of morphological functions. A function called Morphological Dilation is used to fill the holes present in the image followed by Erosion to remove unwanted structures in the image. Disk or Euclidean of radius 1 is used as structuring element. [9]

C. Segmentation

Applied to text, segmentation is the isolation of characters or words. The majority of optical character recognition algorithms segment the words into isolated characters which are recognized individually. Usually this segmentation is performed by isolating each connected component, that is each connected black area. This technique is easy to implement, but problems occur if characters touch or if characters are fragmented and consist of several parts. Therefore, horizontal segmentattion is done carefully as described below.

Segmentation stage being the most important stage, the image is divided into three parts. The topmost part gives the upper area of the descenders, the middle part is the actual overlapping region and the bottom part is the lower part of the ascenders as shown in the figure 5.

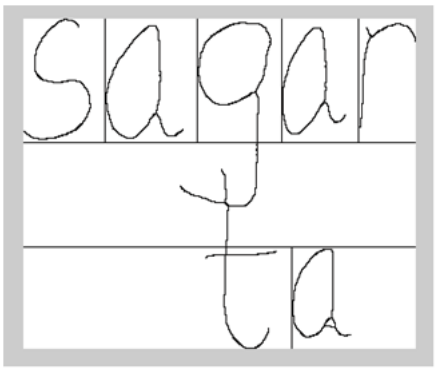

Fig. 5. Horizontal and Vertical Segmentation to Individual Characters

Segmentation [9] [7] is done as follows:

- For detecting upper part, maximum of projection of image pixels on Y axis from topmost line till the mid-line of the image. The line corresponding to this maximum value is labeled.
- Starting from this labeled line till mid-line, the line with less than 2 pixels is the line of segmentation for upper part of image.
- Similarly, it is for the bottom part where the same process starts from bottom last line and goes to the mid line.
- After horizontal segmentation, the image of sequence of characters is segmented vertically into sub-images of individual character. In the process information about number of characters in upper line and lower line is updated.

D. Detection of Overlapping Region

Detection of the overlapped region takes place by the plot called histogram. An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. For our case, we plot a y-histogram as shown in the figure 6.

After segmenting the image to individual characters the region below these characters in upper line are labeled as shown in figure 5.

If the histogram for the labeled overlapping component is non zero everywhere then overlapping or touching is present [1] and image will be sent for processing and recognition of overlapping character. And if the histogram for the labeled overlapping component is zero somewhere then the image is non-overlapping [1] and is sent directly for diagonal feature extraction [2] followed by neural network.

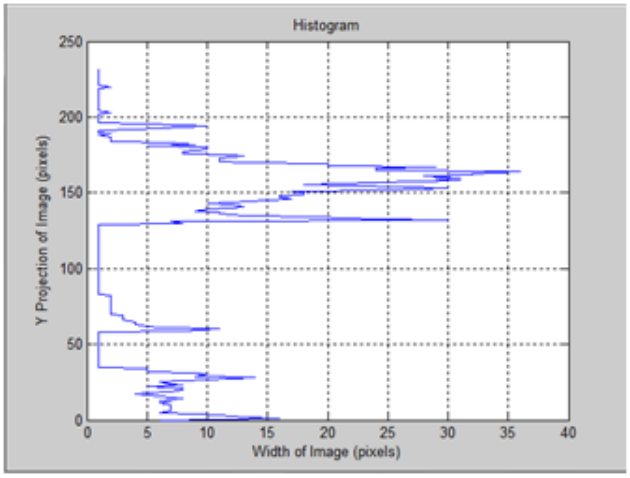

Fig. 6. Histogram or Projection of Image on Y axis

E. Recognition of Overlapped Characters

For recognition the topmost structural part of the descender in the image is compared and differentiated from the upper structural part of the other descenders and the lowermost structural part of the ascender in the image is compared and differentiated from the lower structural part of the other ascenders.

In order to compare and differentiate, we have developed a sequence in which the most obvious characters get detected first by method of elimination. And the sequence is as follows:-

For overlapped characters with descenders:

1) After segmentation, we crop the white spaces surrounding the character and forward it for further processing stages. If the width of the image is less than 15 pixels as shown in the figure 7, the character detected has to be '**j**'.

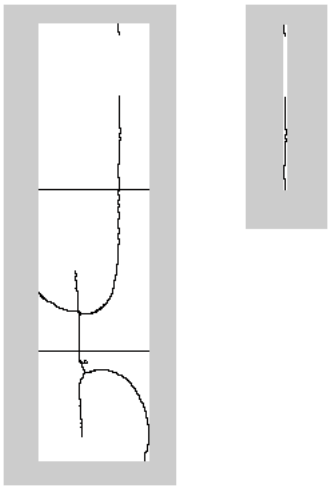

Fig. 7. Detection of J

2) If the width is not less than 15 pixels, then starting from the bottom most line going towards the top row-wise (from left to right), if we encounter a pixel towards the left of the mid-line, then we conclude that the letter is '**p**' as shown in the figure 8. If not, it could either be 'y' or 'g' or 'q'.

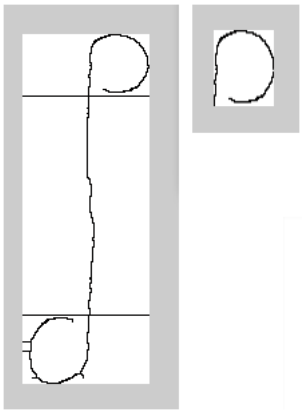

Fig. 8. Detection of P

3) Starting from the top-most line going towards the bottom row-wise (left to right) for first 10 rows, and excluding the first 20 and last 20 columns, we count the number of pixels enclosed in the specified area. If the count is less than 10 pixels, the letter is '**y**'. If not, it could be either 'g' or 'q'.
4) For deciding between 'g' and 'q', we process the overlapping part. The pixels are concentrated on the left hand side for 'g' and on the right hand side for 'q' in the overlapped region. If the number of pixels towards the left of descender of 'g' are more, we detect the character as '**g**'. And if the number of pixels are more on the right hand side of descender of q, we conclude that the character is '**q**'.

For overlapped characters with ascenders, we take input as only the lower segmented image:

1) After segmentation, starting from bottom going towards top wherever average number of pixels in a row is less than 2, that row is labelled. We crop the image from the row labelled till last row and further remove the white spaces surrounding the letter as shown in figure 9. Now if the width of the image is less than 10 pixels, the letter is '**l**' else it can any other.
2) We take the y-projection (histogram) of the image of the character. Now we go row-wise from topmost part of the character towards the bottom and calculate the maxima. Going towards the bottom from maxima, whenever we encounter less than 2 pixels, that particular value of the row is labelled and the entire image is cropped from top till that labelled row. If image width is less than 16% of initial image width as shown in figure 10, we conclude that the letter is '**f**'.
3) Going forward from there, if we take only the image between row labelled in point 2 and point 1. Then we remove the white spaces surrounding that image as shown in the figure 11. Now if the width of image is less than 16% of initial image width then the letter has

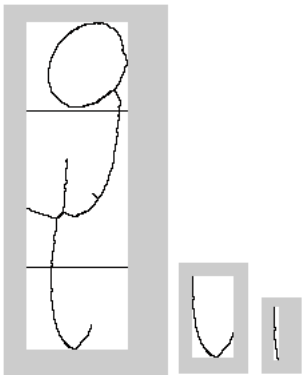

Fig. 9. Detection of L

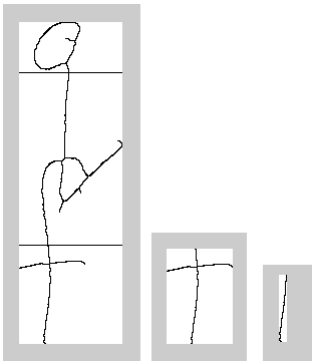

Fig. 10. Detection of F

to be '**t**'.

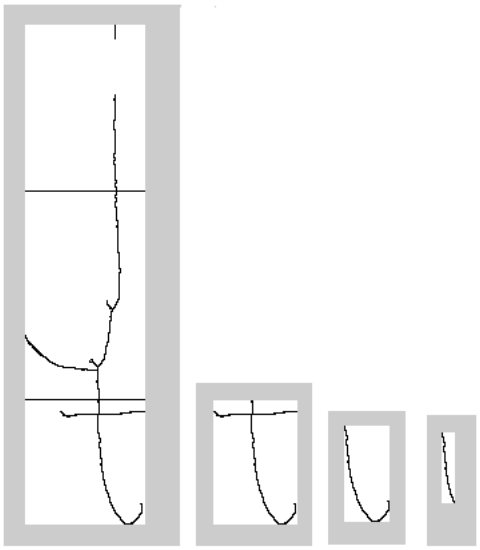

Fig. 11. Detection of T

4) Starting from the topmost row going towards the bottom row-wise from left to right sequentially, if the x co-ordinate of the pixels encountered is greater than the mid value, then we conclude that the letter is '**d**'. If not, then it could be either 'b' or 'h' or 'k'.
5) Now we take into consideration only bottom most 10 rows excluding 20 columns on extreme left as well as extreme right. Count the number of pixels in the enclosed region. If the count is less than 10 pixels, then the letter is '**b**'. If not, then it could be either 'h' or 'k'.
6) Now we take into consideration only first 10 rows excluding 20 columns on extreme left as well as extreme right. Count the pixels in the enclosed region. If the count is less than 10 pixels, then the letter is '**k**', otherwise it is '**h**'.

F. Recognition of Non-Overlapped Characters

Non-overlapped characters are recognized through Decision Theoretic Methods like Neural Network [4] [5] [6], Hidden Markov Model [8], etc. Here we have used Neural Network.

We have used diagonal feature extraction [2] scheme for drawing of features from non-overlapping characters. For the classification stage we use these features extracted. A feed forward back propagation neural network having two hidden layers is used. The architecture used is 69-100-100-26 [2] for classification. The two hidden layers and the output layers uses tan sigmoid as the activation function [4]. The feature vector denoted by F = (f1,f2,f3,...,fd) where d denotes number of training images and each f is having a length of 69 which represents the number of input nodes. The 26 neurons in the output layer correspond to the 26 English alphabets.

The network training parameters [4] [5]:

- Input nodes : 69
- Hidden nodes : 100 each
- Output nodes : 26
- Training Algorithm : Scaled conjugate gradient back propagation
- Performance function : Mean Square Error
- Training goal achieved : 0.000003
- Training epochs : 5000

IV. Experimental Results and Analysis

The recognition system has been implemented. This recognition of overlapped characters is accomplished by fuzzy reasoning. The detection and recognition of the non overlapped is done by the Feed Forward Back Propagation Neural Network.

Descender	Successful Recognition	False Recognition	No Recognition
g	90%	8%	2%
p	92%	8%	0%
q	90%	9%	1%
j	93%	7%	0%
y	88%	20%	2%

Ascender	Successful Recognition	False Recognition	No Recognition
b	88%	11%	1%
d	90%	9%	1%
f	83%	15%	2%
h	70%	29%	1%
t	85%	15%	0%
l	92%	7%	1%
k	70%	28%	2%

A. Successful Cases

The tesing algorithm has been applied to all standard 35 cases of overlapping characters in various handwriting samples as shown in the figures 12. An accuracy of 85.92% has been achieved in all the cases. For further testing, non overlapped characters were also tested with the same algorithm. The accuracy achieved here is very high, 99%

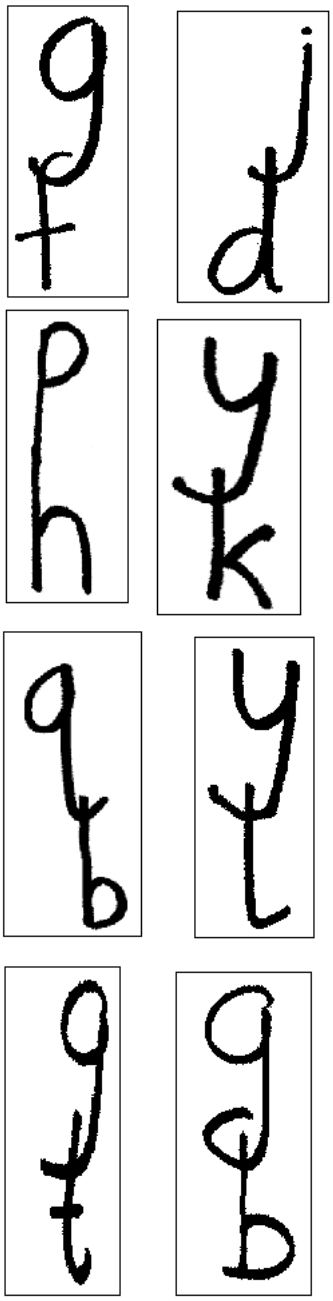

Fig. 12. Few Successful Cases

B. Failed Cases

After successful training and testing of the algorithm, we have encountered the following flaws which results in the following failed cases:

Fig. 13. Failed case 1

1) Due to the occurance of overly large and disproportionate ascender as compared with the descender in the figure 13, the descender 't' exceeds the threshold which is decided on the basis of image height.

Fig. 14. Failed case 2

2) As shown in the figure 14, the lower part of the ascender which is processed for the detection, is slanted. Hence, the result of detection is incorrect.

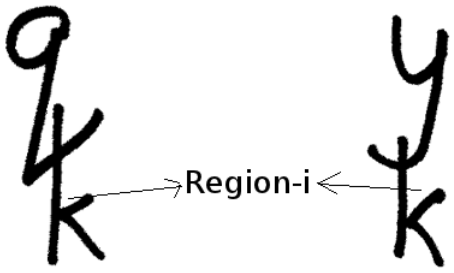

Fig. 15. Failed case 3

3) As labelled in the figure 15, the region-i of the image on the left hand side has less white space and hence more pixel concentration. However, the same region-i of the right hand side of image has more white space and hence less pixel concentration. Due to this, the left side of the image will be a failed case and will always detect it as

'h' and not 'k' and the right side of the image will be a successful case.

V. Conclusion

When compared with other algorithms in the same area of research, our method implemented here works with high accuracy in almost all cases which include intersections of loop and instances of multiple criss-crossings. This is mainly because our algorithm, unlike others, is focussed on the processing of the non-overlapped part rather than the overlapped part.

The method implemented functions for the following conditions:

- This algorithm is valid only for non cursive english lower case alphabets
- The algorithm requires that the height of the ascenders as well as the descenders to be proportionally same
- For extermely illegible handwriting, the accuracy achieved by the algorithm is very less

The described method for separation of overlapped characters between adjacent lines of handwritten documents has demonstrated effective results in all cases of the predefined ascenders and descenders database described. The problem of detecting and separating overlapped characters in general and outside the predefined database is still the subject of future research.

References

[1] Kalyan Takru and Graham Leedham, "Separation of touching and overlapping words in adjacent lines of handwritten text", Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), 2002.

[2] J.Pradeep, E.Srinivasan and S.Himavathi, "Diagonal based feature extraction for handwritten character recognition system using neural network", 978-1-4244-8679-3/11, 2011.

[3] Dan Ciresan and Dan Pescaru, "Off-line Recognition of Handwritten Numeral Strings Composed from Two-digits Partially Overlapped Using Convolutional Neural Networks", IEEE 978-1-4244-2673-7/08, 2008

[4] Amit Choudhary, Rahul Rishi, Savita Ahlawa and Vijaypal Singh Dhaka, "Performance Analysis of Feed Forward MLP with various Activation Functions for Handwritten Numerals Recognition", 978-1-4244-5586-7/10, IEEE, 2010

[5] Rokus Arnold and Poth Miklos, "Character Recognition Using Neural Networks", CINTI 2010 11th IEEE International Symposium on Computational Intelligence and Informatics, 18-20 November, 2010

[6] J.Pradeep, E.Srinivasan and S.Himavathi, "Neural Network based Handwritten Character Recognition system without feature extraction", International Conference on Computer, Communication and Electrical Technology,18-19 March, 2011

[7] R. V. Kulkarni and P.N. Vasambekar, "An Overview of Segmentation Techniques for Handwritten Connected Digits", 2010 International Conference on Signal and Image Processing, 2010.

[8] Salvador Espana-Boquera, Maria Jose Castro-Bleda, Jorge Gorbe-Moya, and Francisco Zamora-Martinez,"Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models",IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 33, NO. 4, APRIL 2011

[9] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing (3rd Edition)", ISBN-13: 978-0131687288, Publication Date: August 31, 2007.

BIOGRAPHIES

Raj Sheth is an undergraduate student in Electronics and Telecommunication Department at Sardar Patel Institute of Technology, India. His areas of interest are Embedded systems, Real Time Operating Systems and Networking.
Email id: rajsheth24@gmail.com

Ketan Patil is an undergraduate student in Electronics and Telecommunication Department at Sardar Patel Institute of Technology, India. His areas of interest are Telecommunication and Networking.
Email id: ketan.federer@gmail.com

Niharika Thakur is an undergraduate student in Electronics and Telecommunication Department at Sardar Patel Institute of Technology, India. Her areas of interest are Wireless Networks.
Email id: niharika.thakur11@gmail.com

K. T. Talele is an Associate Professor in Sardar Patel Institute of Technology, Mumbai, India. His area of interest includes DSP, Image Processing and Computer Vision and Graphics. He has published 45 papers in various refereed international journals and conferences.
Email id: kttalele@spit.ac.in